Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования  
 «Севастопольский государственный университет»

**Отчёт**

по лабораторной работе

**Исследование однослойного персептрона**

Дисциплина «Основы нейронных сетей»

Выполнил студент гр. ГНКЗ-8

Мовенко К.М.

Вариант − 5

**Севастополь**

**2023**

# ЦЕЛЬ РАБОТЫ

Углубление теоретических знаний в области архитектуры нейронных сетей с пороговыми активационными функциями, исследование свойств однослойного персептрона и правила его обучения, приобретение практических навыков обучения и моделирования однослойной сети при решении простых задач классификации.

# ЗАДАНИЕ

Таблица 1 – Вариант задания №5 к п.1

|  |  |
| --- | --- |
| Матрица **P** | Вектор **T** |
|  |  |

Таблица 2 – Вариант задания №5 к п.2

|  |  |  |  |
| --- | --- | --- | --- |
| Класс 1  **P(:,1), P(:,2),** | Класс 2  **P(:,3), P(:,4),** | Класс 3  **P(:,5), P(:,6),** | Класс 4  **P(:,7), P(:,8),** |
| [2; 1],  [2; 2] | [-2; 1],  [-2; 2] | [-2; 0],  [-2; -1] | [2; 0],  [2; -1] |

1. Для заданной матрицы входных данных **P** и заданного вектора выходных значений **T** (Таблица 1) разработать простой персептрон, решающий задачу классификации 2-х классов. Решение найти с помощью графического построения границы решения и вычисления весов и смещения вручную. Протестировать с помощью компьютера полученное решение для всех входных векторов (столбцов матрицы **P**);
2. Даны четыре класса, каждый из которых представлен 2-мя точками (столбцами матрицы **P**), указанными в Таблице 2. Необходимо:

* разработать структурную схему персептрона, распознающего эти 4 класса;
* выполнить предварительный анализ задачи, изобразив все точки четырёх классов и построив графически возможные границы решений персептрона;
* анализируя графическую границу решений, задать допустимые целевые значения выходов персептрона (определить матрицу **T**) для всех входных точек, представленных столбцами матрицы данных **P**;
* изучить функцию ann\_PERCEPTRON и, используя её, обучить персептрон правильному распознаванию входных классов;
* написать программу, которая:
  + отображает диаграмму размещения входных точек из **P** на плоскости с координатами (*,*);
  + обучает персептрон;
  + накладывает на диаграмму входных точек границы решения после обучения персептрона;
  + выполняет тестирование полученного решения для всех заданных входных данных, а также для дополнительно выбранных точек из областей принадлежности входных классов;

1. Выполнить анализ полученных результатов, обратив внимание на то, что при каждом новом запуске функции обучения границы решения меняются. Сравнить результаты нескольких запусков программы. Определить, сохранит ли программа работоспособность, если матрицу целевых значений выходов T формировать произвольным образом (например, используя в качестве целевого выходного вектора номер его класса в двузначном двоичном коде);

# ЛИСТИНГИ ПРОГРАММ

Листинг 1 – Тестирование персептрона

clear;

w = [2 2]; *// вычисленный вектор весов*

b = -3; *// вычисленное смещение*

*// заданная матрица входов*

P = [0 1 1 0 ;

0 0 1 1];

*// ожидаемые значения на выходе*

T = [0 0 1 0];

*// моделирование персепетрона*

a = ann\_PERCEPTRON\_run(P, w, b);

disp("Ожидаемые значения: ");

disp(T)

disp("Полученные значения: ")

disp(a)

*// проверка решения*

if isequal(a, T) then

disp("Решение корректное")

else

disp("Решение не корректное")

end

Листинг 2 – Обучение двухслойного персептрона для определения классов

clear *// очистка рабочего пространства*

K = 4; *// количество классов*

N = 2; *// количество точек в классе*

S = 2; *// число нейронов в слое*

*// входные значения*

P=[2 1 -2 0 1 2 -1 -2;

1 2 -2 -1 -2 -1 0 1];

*// желаемые выходные значения*

T=[0 0 1 1 1 1 0 0;

0 0 1 1 0 0 1 1];

*// вызов функции обучения*

[w,b] = ann\_PERCEPTRON(P,T);

*// построение диаграммы расположения входных точек*

clf;

j = -3; *// код отображаемого символа*

for i = 1:K\*N *// цикл для всех точек данных из P*

plot2d(P(1,i), P(2,i), j);

if modulo(i,2) == 0

j = j-1; *// изменение символа отображения*

end

end

xgrid;

xtitle("Границы решений", "p1", "p2");

mtlb\_hold('on') *// запрет очистки граф. окна*

*// отображение на графике границ решений*

p1 = min(P) - 0.1:0.1:max(P) + 0.1;

len = length(p1);

p2 = zeros(S, len);

for i = 1:S

p2(i,:) = -w(i,1) / w(i,2) .\* p1 - b(i) ./ w(i,2); *// уравнение прямой*

plot(p1, p2(i,:));

end

*// тестирование персептрона*

a = ann\_PERCEPTRON\_run(P,w,b) *// на входных данных*

disp(a)

*// дополнительные точки входных классов*

P=[1.1 1.7 0 -2.5 1.3 2.5 -1.5 -2.3;

1.9 2.2 -1.3 -1.5 -2.1 -1.1 0.3 0.7];

a = ann\_PERCEPTRON\_run(P,w,b) *// для доп. точек*

disp(a)

# ХОД работы

Необходимо построить простой персептрон, удовлетворяющий следующим требованиям:

Для нахождения параметров персептрона аналитическим методом было выполнено графическое построение границы решений (Рисунок 1).

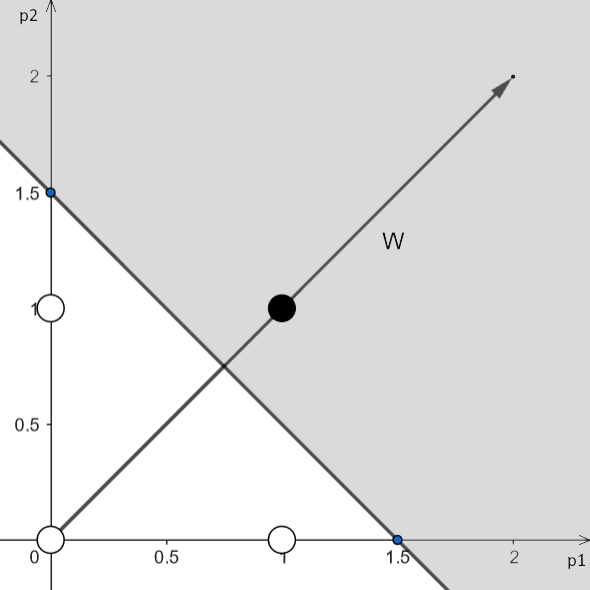


Рисунок 1 – Граница решения персептрона

Граница была проведена через точки (1,5; 0) и (0; 1,5). Вектор весов был проведён перпендикулярно ей с координатами (2; 2).

Для нахождения смещения была взята точка на границе (1,5; 0).

Персептрон с найденными параметрами был протестирован в сценарии Scilab (Листинг 1). В результате программа подтвердила верность полученного решения (Рисунок 2).

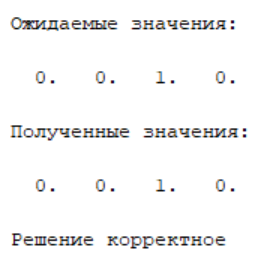


Рисунок 2 – Результат тестирования персептрона

Затем были рассмотрены классы п.2. Исходные значения не удовлетворяют требованиям к заданию, т.к. одна и та же точка не может лежать в двух классах одновременно. Матрица P была частично преобразована:

Для указанных точек были построены графически возможные границы решений (Рисунок 3).

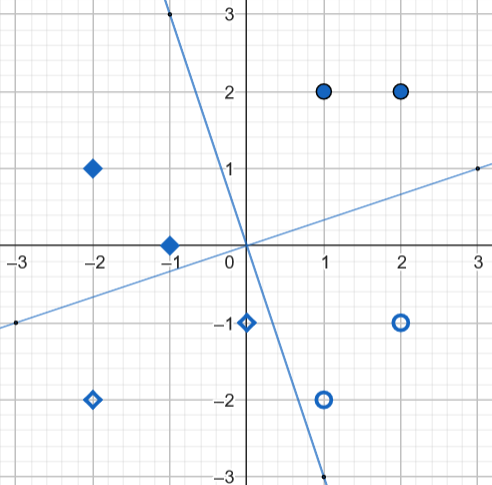


Рисунок 3 – Границы решений для классов задачи 2

Для каждого класса были заданы ожидаемые выходы:

* класс 1: t = [0; 0] (закрашенные точки);
* класс 2: t = [1; 1] (не закрашенные ромбы);
* класс 3: t = [1; 0] (не закрашенные точки);
* класс 4: t = [0; 1] (закрашенные ромбы);

Значения входных и выходных значений были забиты в написанную программу (Листинг 2). По ним нейросеть была обучена, был выведен график с входными точками и найденными программой границами решений (Рисунок 4).

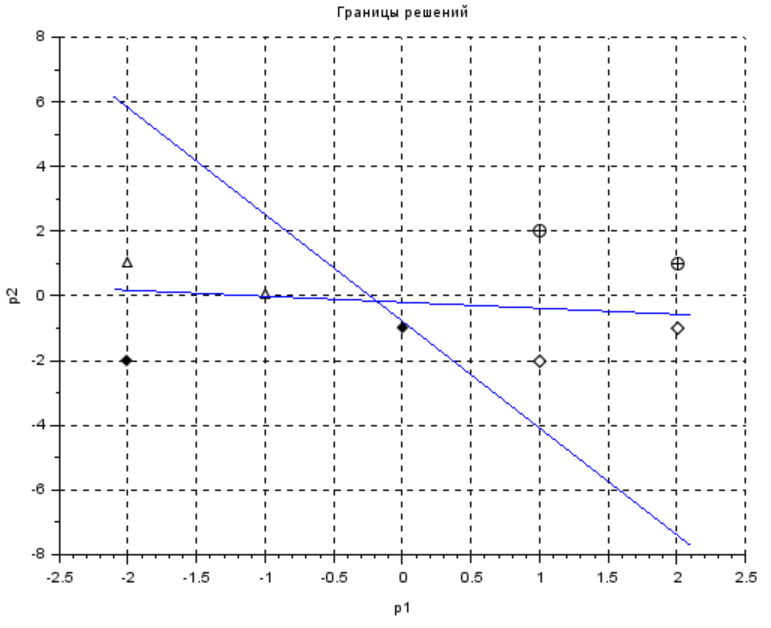


Рисунок 4 − Диаграмма расположения входных точек и границы решений

Для тестирования результатов обучения персептрона была выведена матрица выходных значений. Также на вход персептрона были поданы альтернативные значения, соответствующие тем же классам, что и изначальные. Выходные значения не изменились (Рисунок 5), следовательно обучение было выполнено верно.

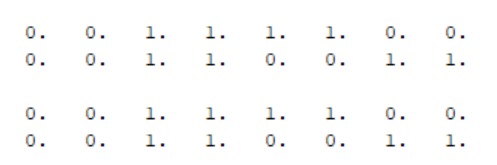


Рисунок 5 – Тестирование персептрона на различных входных значениях

Затем программа была запущена повторно. Значения на выходе всё также удовлетворяют ожиданиям, однако границы решений при на диаграмме изменились (Рисунок 6). Это справедливо для всех повторных запусках.

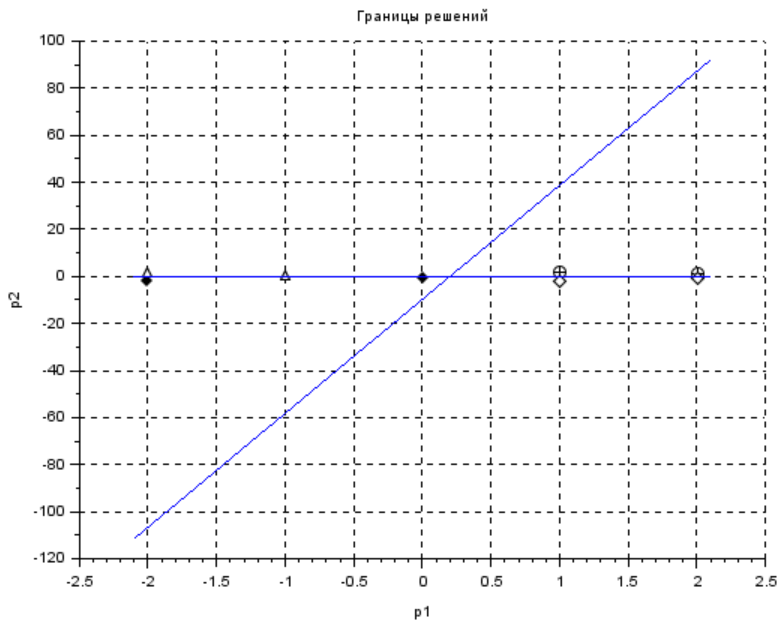


Рисунок 6 – Диаграмма границ решений при повторном запуске

Также программа была протестирована для произвольных значений матрицы T. Если в качестве выходного вектора использовать двоичное представление класса обучение пройдёт успешно, однако это лишь случайность. При произвольных значениях T, указанных ниже, обучение невозможно, программа не завершается.

# ВЫВОД

В ходе работы было проведено исследование простейших однослойных персептронов на одном и двух нейронах. Освоены графический и программный методы нахождения границ решений персептрона. Было проведено обучение персептронов на наборах входных и ожидаемых данных.

Также была разработана схема персептрона, распознающего 4 класса. Для этого был выполнен предварительный анализ задачи – была рассмотрена диаграмма с точками входных значений.

При тестировании полученного персептрона было замечено, что при каждом испытании границы решений различны. Это связано с тем, что функция обучения персептрона каждый раз начинает работу с различных значений, что отображается на результате.

Также было замечено, что матрицу ожидаемых на выходе значений нельзя формировать производным образом. Это подтверждает значимость предварительного анализа схемы перед формированием обучающих данных.